IN THE SPECIFICATION

Please amend the specification as follows:

On page 2, please amend the paragraph that begins at line 17 as follows:

Fig. 2a [[2]] illustrates a method of decoupling store address and data in a multiprocessor system according to one example embodiment of the present invention;

Fig. 2b illustrates a method of decoupling store address and data in a multiprocessor system according to another example embodiment of the present invention;

On page 5, please amend the paragraph that begins at line 24 as follows:

It is desirable to split the write operations up into two parts-- a write address request and a write data request--and send each out to memory system 16 separately. One embodiment of such a method is shown in Fig. 2a [[2]]. In the embodiment shown in Fig. 2a [[2]], write address requests are sent to memory 16 at 50, where they are held in the memory system at 52, either by changing the state of the associated cache lines in a cache, or by saving them in some structure. The purpose of the write address request is to provide ordering of the write request with subsequent requests. Once the write address request has been sent out to the memory system, requests from other processors that are required to be ordered after the write can be sent out to the memory system, even though the data for the write request has not yet been produced.

On page 7, please amend the paragraph that begins at line 20 as follows:

In one example embodiment, even though the write data and write address are sent at different times, they are received in instruction order at memory 16. In such an embodiment, you don't have to send an identifier associating an address with its associated data. Instead, the association is implied by the ordering. Such an example embodiment is illustrated in Fig. 2b. At 70, a write request address for a memory write is generated in one of a plurality of processors, wherein the write request address points to a memory location in a shared memory coupled to the processors across a network. At 71, a write request is transferred from the processor to the shared memory, wherein the write request includes the write request address. At 72, the write

request address is noted in the shared memory. At 73, memory ordering is enforced in subsequent load and store requests to the write request address until the write data associated with the write request is written into the shared memory. At 74, when the corresponding write data becomes available, the corresponding write data is transferred to the shared memory in instruction order across the network without the write request address. At 75, the write request address is paired, within the shared memory, with the separately transferred corresponding write data. Then, at 76, the write data is stored into the shared memory as a function of the write request address.